Assignment 2: Coding Basics

Aileen Lavelle

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on coding basics.

Directions

1. Rename this file `<FirstLast>_A02_CodingBasics.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.
6. After Knitting, submit the completed exercise (PDF file) to Sakai.

Basics Day 1

1. Generate a sequence of numbers from one to 100, increasing by fours. Assign this sequence a name.
2. Compute the mean and median of this sequence.
3. Ask R to determine whether the mean is greater than the median.
4. Insert comments in your code to describe what you are doing.

```
#1.
seq(1,100) #use the seq function to see the vector of 1-100
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100
```

```
#2.
#setting "a" as 1-100
a <- seq(1,100)
#calculate mean of "a"
mean(a)
```

```
## [1] 50.5
```

```r
mean_a <- mean(a)
mean_a
```

```
## [1] 50.5
```

```r
#calculate median of "a"
median(a)
```

```
## [1] 50.5
```

```r
median_a <- median(a)
median_a
```

```
## [1] 50.5
```

```r
#3.
#seeing if mean is larger than median
mean_a > median_a
```

```
## [1] FALSE
```

Basics Day 2

5. Create a series of vectors, each with four components, consisting of (a) names of students, (b) test scores out of a total 100 points, and (c) whether or not they have passed the test (TRUE or FALSE) with a passing grade of 50.
6. Label each vector with a comment on what type of vector it is.
7. Combine each of the vectors into a data frame. Assign the data frame an informative name.
8. Label the columns of your data frame with informative titles.

```r
#name vector of test scores as "student_scores"
student_scores <- c(89, 43, 100, 25)
student_scores
```

```
## [1]  89  43 100  25
```

```r
#conditional statement on pass/fail
Fail <- ifelse(student_scores<50, TRUE, FALSE)
Fail
```

```
## [1] FALSE  TRUE FALSE  TRUE
```

```r
#create vector of student names
student_names <-c("Maeve", "Laura", "Ally", "Aileen")
student_names
```

```
## [1] "Maeve"  "Laura"  "Ally"   "Aileen"
```

```r
#convert vector of student names to a data frame
student_names <-as.data.frame(student_names)
student_names
```

```
##   student_names
## 1         Maeve
## 2         Laura
## 3          Ally
## 4        Aileen
```

```r
# add rows using cbind()
df <-cbind(student_names,student_scores,Fail)
df
```

```
##   student_names student_scores  Fail
## 1         Maeve             89 FALSE
## 2         Laura             43  TRUE
## 3          Ally            100 FALSE
## 4        Aileen             25  TRUE
```

```r
#Checking to make sure df is now a dataframe
is.data.frame(df)
```

```
## [1] TRUE
```

9. QUESTION: How is this data frame different from a matrix?

   Answer: A data frame can contain different data types, ie. characters, numbers, factors and times. A matrix can only contain a single type of data.

10. Create a function with an if/else statement. Your function should take a **vector** of test scores and print (not return) whether a given test score is a passing grade of 50 or above (TRUE or FALSE). You will need to choose either the `if` and `else` statements or the `ifelse` statement.

11. Apply your function to the vector with test scores that you created in number 5.

```r
#ifelse function
Fail <- ifelse(student_scores<50, TRUE, FALSE)
print(Fail)
```

```
## [1] FALSE  TRUE FALSE  TRUE
```

```r
#seperate if/else statement
Failed_Students <- function(student_scores) {
  if(x < 50) {
      (TRUE)
  }
  else if (x > 50) {
      (FALSE)
  }
```

```
  else {
    x
  }
}
print(Failed_Students)
```

```
## function(student_scores) {
##   if(x < 50) {
##       (TRUE)
##   }
##   else if (x > 50) {
##       (FALSE)
##   }
##   else {
##     x
##   }
## }
```

12. QUESTION: Which option of `if` and `else` vs. `ifelse` worked? Why?

    Answer: The option of ifelse works. The seperated if and else will display all of the command and output.